American Journal of Botany and Bioengineering
https://biojournals.us/index.php/AJBB
ISSN: 2997-9331

Measuring Temperature and Humidity Using Arduino

Hussien Essam Adnan Jawad, Abdulhameed Dhari Hameed Rijah, Abdalbacet Azez Ali Sanad, Ahmed Adil Naim Alwan
Department of Biomedical Engineering, University of Technology, Iraq

Asaad Muzhr Mathbob Mankhi
Department of Medical Instrumentation Engineering, Middle Technical University, Iraq

Received: 2025 19, Jan
Accepted: 2025 28, Feb
Published: 2025 04, Mar

Annotation: Technology plays a crucial role in modern life, especially in home automation and healthcare applications. This study presents a system for measuring temperature and humidity using an Arduino-based setup, which is beneficial for both environmental monitoring and health management. The system utilizes a DHT11 sensor to collect real-time data, displayed on an LCD screen. The hardware includes an Arduino Uno board, jumper wires, LEDs, resistors, a buzzer, and a breadboard for easy assembly. The Arduino IDE ensures seamless integration of components. This device not only enhances indoor comfort and energy efficiency but also contributes to health safety by maintaining optimal environmental conditions, reducing risks of respiratory issues and heat stress. A buzzer provides alerts when temperature or humidity exceeds set thresholds, while LEDs indicate environmental status. This project demonstrates the practical use of embedded systems in climate control and healthcare, offering an affordable and efficient solution. Future enhancements may include wireless connectivity and data logging for improved usability.

Keywords: Temperature measurement. DHT11 sensor. Climate change detection. Humidity measurement. Arduino Uno R3Sound and light alert system.

1. Introduction

The term "technology" has become widely used in the twenty-first century. Technology has many and varied uses, including Bluetooth headsets, tablets, televisions, and more. Today, it is impossible for our homes to be devoid of electronic (technological) devices such as cell phones, home computers (PCs), iPads, laptops, medical devices, and other devices. Today, measuring the temperature of a substance or thing has become extremely easy, such as measuring heart rate, gas levels, water levels, light intensity, temperature, humidity, and more. In this project, a device was created to measure the temperature and humidity of a room and then display them on an LCD screen, using a world-famous electronic board called Arduino. (Arduino), in addition to peripheral devices that are connected to the Arduino board.

2. Project Objectives

It is not uncommon for family members to have varying preferences regarding room temperatures. The comfort experienced by one individual at a temperature of 20 degrees Celsius may differ significantly from that of their partner, siblings, or parents. This highlights the importance of accurately measuring room temperature. Attaining a suitable temperature that aligns with your personal comfort is essential for several reasons. Prioritizing your ideal temperature is vital, as excessive warmth can hinder concentration, while being too cold may elevate the risk of illnesses such as colds, heart attacks, and pneumonia. Additionally, financial considerations play a role; heating systems, including central heating, can incur substantial costs. Therefore, it is prudent to be as economical as possible, which can also contribute to reducing carbon emissions and protecting the environment. Generally, the optimal temperature for sleep is between 16 and 18 degrees Celsius. Humidity, defined as the concentration of water vapor in the air, is another critical factor in our daily lives, particularly in relation to weather predictions. Its significance is especially pronounced concerning human health, as high humidity levels can lead to heat stress and respiratory issues. A comfortable humidity level typically falls between 40% and 60% when the room temperature is approximately 22 degrees Celsius.

3. Components of the Project

3.1. Arduino UNO R3

This board is recognized as the first and most popular Arduino model, renowned for its user-friendliness. It consists of a compact electronic circuit that offers ports for the direct connection of electronic components to the microcontroller through digital input and output ports. A key characteristic of this design is that the microcontroller chip is not permanently affixed to the board; instead, it is mounted on a removable stand. This aspect makes it an ideal option for novices. In the event that the chip becomes damaged during a project, the board can be easily repaired by substituting the microcontroller chip with another of the same type.

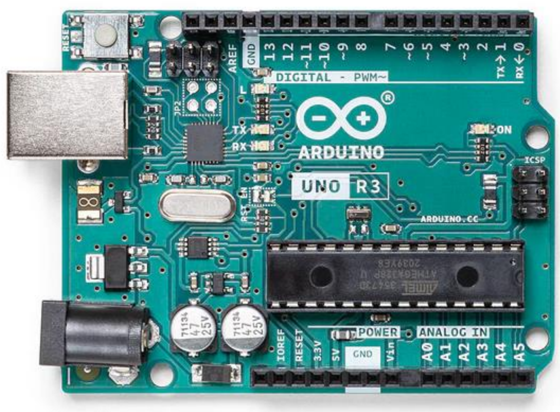


Fig (1); Arduino Uno.

The Arduino is programmed using a USB cable, type A-B, which connects the Arduino to a personal computer. It can also be connected to a mobile phone using an OT cable.

Fig (2): USB cable.

3.2. Temperature and Humidity Sensor

The DHT11 temperature and humidity sensor was used, as shown in Figure (3) and Figure (4).

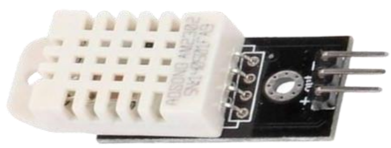


Fig (3): DHT22

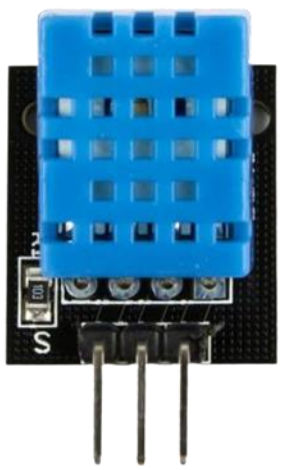


Fig (4): DHT11

The DHT11 is part of a range of humidity and temperature sensors that are both affordable and of satisfactory quality. The DHT22, illustrated in Figure (3), is the other sensor in this lineup. Both devices function as relative humidity (RH) sensors, capable of measuring both temperature and humidity levels. The digital output from these sensors can be processed by any microcontroller or processor for further analysis. The DHT11 specifically measures relative humidity values ranging from 0% to 90% and temperature values from 0°C to 50°C, with data being updated every second during its 1-second sampling period.

3.3. Jumper Wires

Many connecting wires of two types were used (male-male) as shown in Figure (5) and (female-male) as shown in Figure (6).

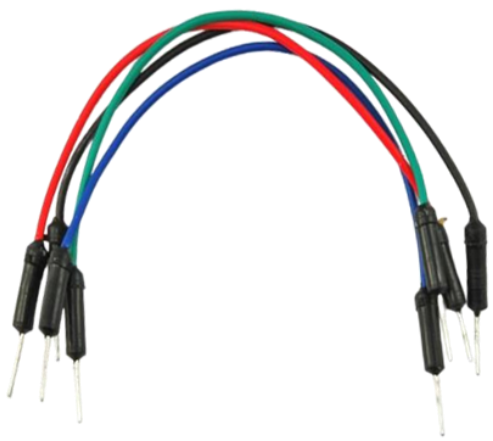


Fig (5): male-male.

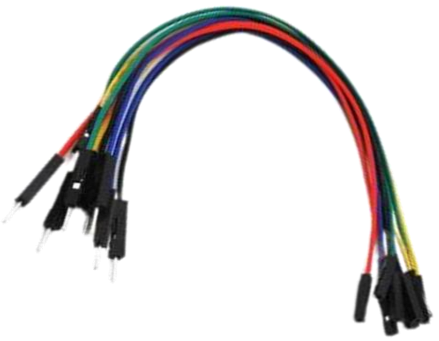


Fig (6): female-male.

Electrical wiring is very important for connecting the Arduino to the breadboard, sensors or LCD screens.

3.4. Light Emitting Diodes

Two LEDs were used to give the signal.

Fig (7): LEDs.

A light-emitting diode, commonly referred to as an LED, is a type of light source constructed from semiconductor materials that produce light when an electric current flows through them. LEDs are regarded as more energy-efficient compared to traditional electric lamps in terms of power usage.

3.5. Bread Boards

The breadboard is utilized alongside the Arduino or Raspberry Pi, as well as in various electronic circuits. It allows for the creation of connections and circuits without the need for soldering, making it easier to modify and disassemble components.

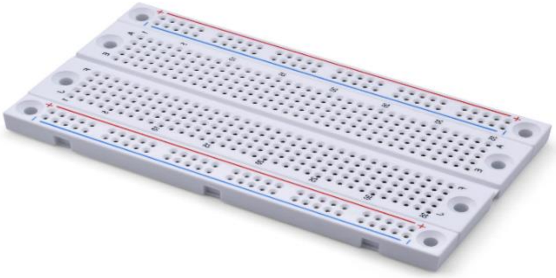


Fig (8): Bread Boards.

3.6. Electrical Resistances

Two resistors were utilized: one with a resistance of 220 ohms (220 Ω) and another with a resistance of 10 kilo ohms (10 kΩ). Electrical resistance is a fundamental characteristic of metallic conductors within electrical circuits. It is defined as the capacity of a material to hinder the flow of electric current. Essentially, it represents the material's opposition to the movement of electric current (electrons) through it.

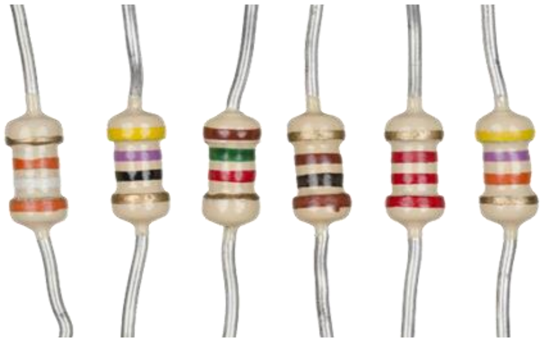


Fig (9): Electrical Resistances

3.7. Buzzer

An electric buzzer is a device that transforms electrical energy into sound that can be heard. It serves as an audible alert in various applications, including automobiles and microwave ovens. Electric buzzers come in various types. A standard buzzer typically functions on a voltage range of 6 to 12 volts and operates with a direct current of around 25 milliamps.

Fig (10): Buzzer.

3.8. LCD Screen

A 20x4 LCD was utilized in conjunction with an I2C circuit to simplify the connection process. LCDs are displays, available in various sizes, that many people recognize from devices like alarm clocks, digital clocks, calculators, and others. The variety of electronic devices is vast, and the integration of Arduino with an LCD and open-source hardware enhances the possibilities. This display is compatible with any device or microcontroller, including Arduino project boards; however, these boards necessitate specific connectors or pins to link the electronic board to the LCD.

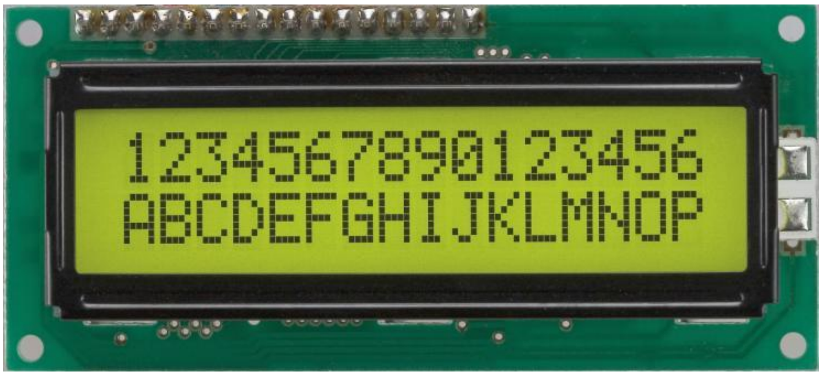


Fig (11): LCD Screen.

4. Connecting the project components

4.1. Arduino Connections

➢ Breadboard:

✓ Connect the Arduino's GND pin to the breadboard (black wire).

✓ Connect the Arduino's VCC pin to the breadboard (red wire).

➢ LCD screen:

✓ To power the display, connect the VCC pin (red wire) of the LCD's I2C circuit to the breadboard with the red line.

✓ Connect the GND pin (black wire) to the GND pin of the Arduino.

✓ To transmit data to the display, connect the orange wire from the SDA pin of the I2C circuit connected to the LCD to pin A4 on the Arduino.

✓ Connect the yellow wire from the SCL pin of the I2C circuit connected to the LCD to pin 5A on the Arduino.

➢ DHT11:

✓ Place the DHT11 sensor on the breadboard.

✓ To power the sensor, connect the sensor's VCC (red wire) to the red pin on the breadboard and the GND (black wire) to the blue pin on the breadboard.

✓ Connect the DATA pin (green wire) to pin D2 on the Arduino. Connect a 10 kΩ resistor with one end connected to the DATA pin and the other end to VCC on the breadboard (red line).

➢ Buzzer:

Connect the black wire to GND on the Arduino board, and the other wire (red) to pin 7D on the Arduino board.

➢ LED :

Connect the red LED (purple wire) and the green LED (grey wire). In the first pole, place two 220-ohm (220 kΩ) resistors, the first end of which is connected to the straight pole and the second end to GND or to the breadboard with the blue line. Then connect the second end to 9D (grey wire) and D8 (purple wire).

5. the Schema of the Project

Figure 12 shows the connection of the project components, including sensors, wires, LEDs, resistors, and buzzer, and their connection to the Arduino board to implement it.

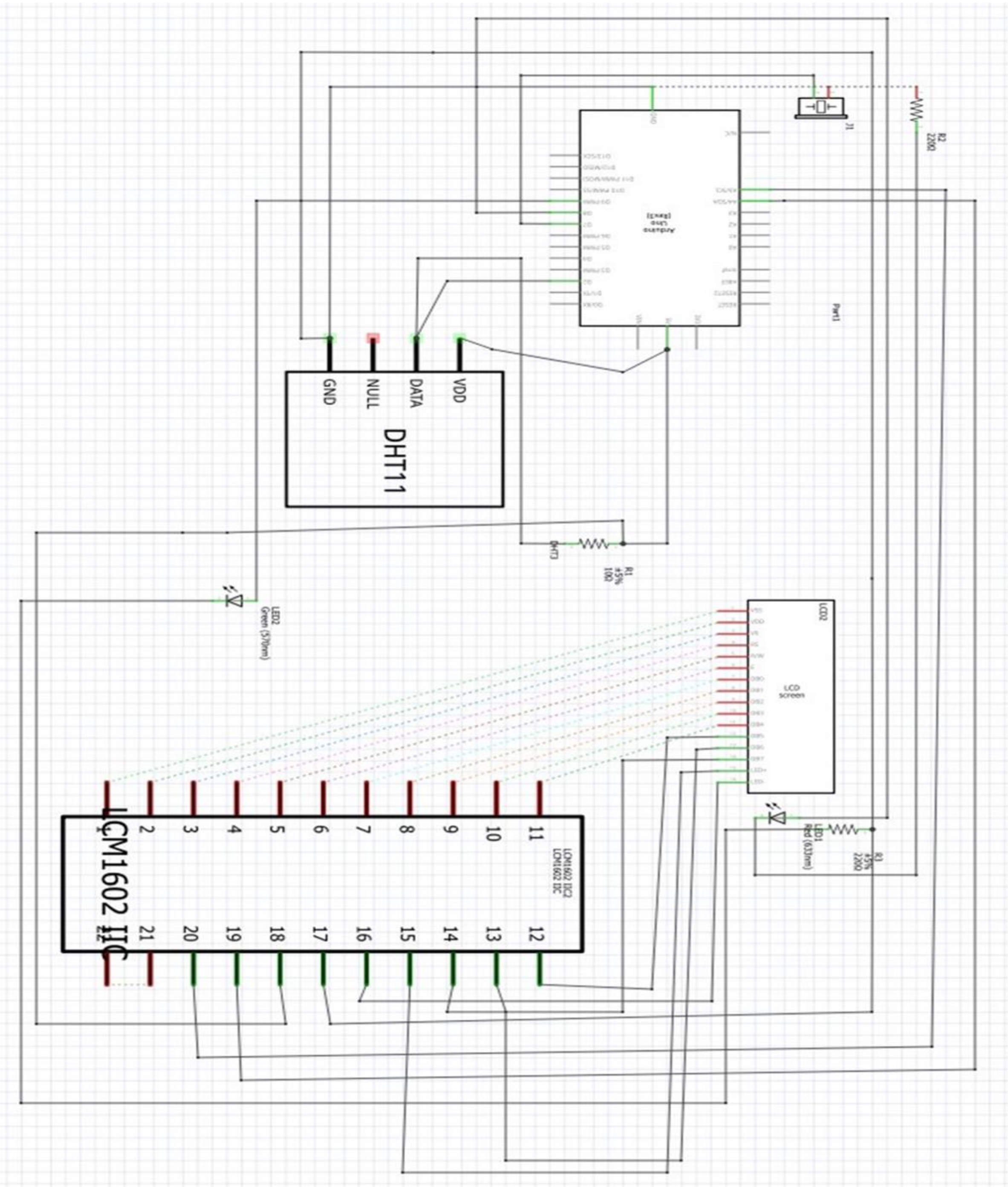


Fig (12): the Schema of the Project.

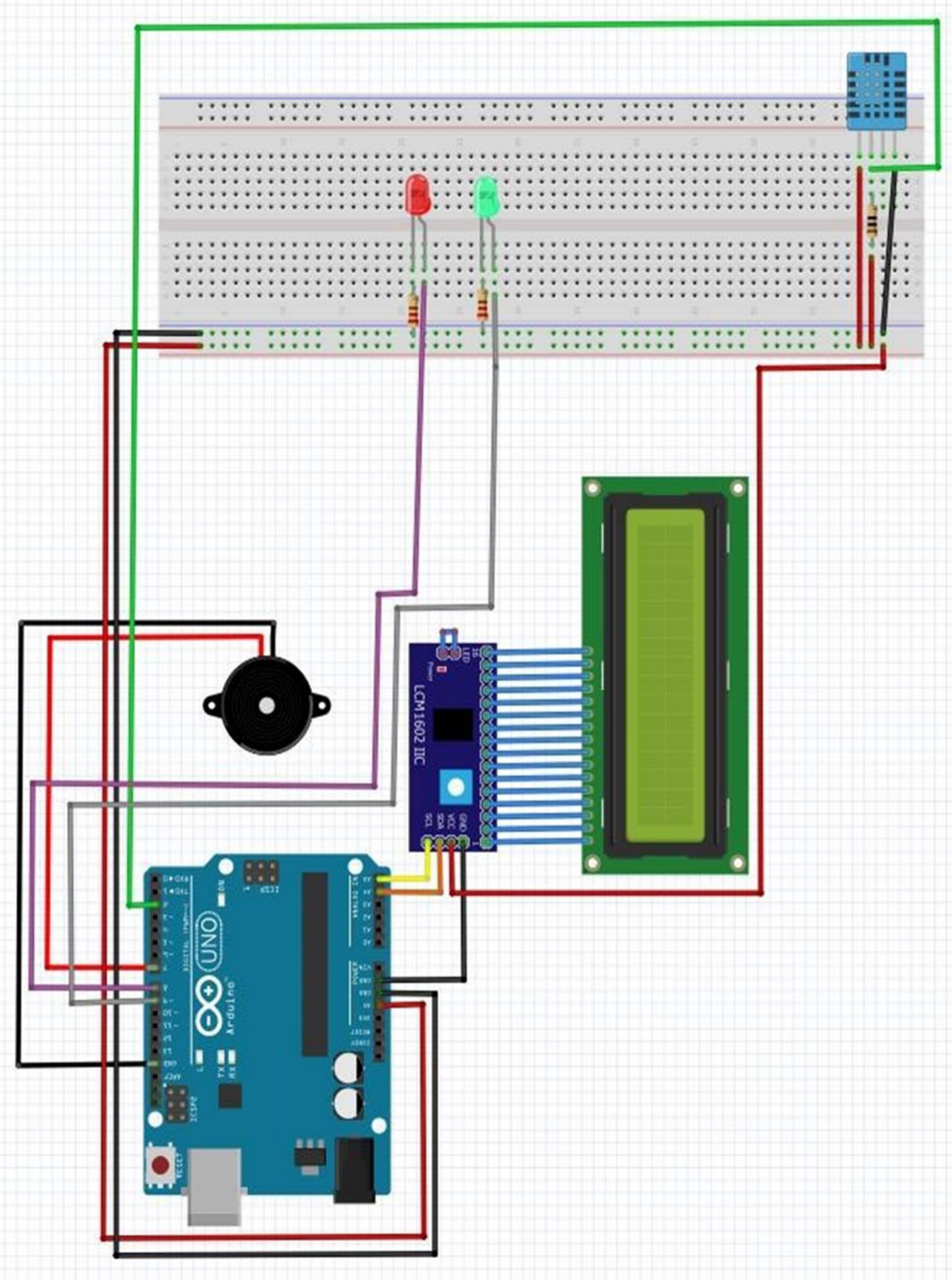


Fig (13): Linking project components.

6.the Algorithm of the Project:

6.1 Flowchart

The flowchart in the figure (14) represents a process for measuring the temperature and humidity in a room using an Arduino-based system and sensors. The flowchart can be analyzed as follows:

1. Start: The process begins when the system is powered on.
2. Measure Temperature and Humidity: The system collects environmental data using a DHT11 sensor or a similar device.

3. Temperature Comparison: The system checks if the temperature is greater than 26°C.

If the temperature is above 26°C:

A buzzer is activated to sound an alert.

A green LED turns on to indicate high temperature.

If the temperature is 26°C or below:

A red LED turns on to indicate that the temperature is within the normal range.

4. Display Data: The temperature and humidity values are displayed on an LCD screen for user monitoring.
5. Refresh Check: The system determines whether to refresh the data and continue measurement. If the answer is "Yes," the process repeats; otherwise, it stops.
6. End: The process ends if no further updates are needed.

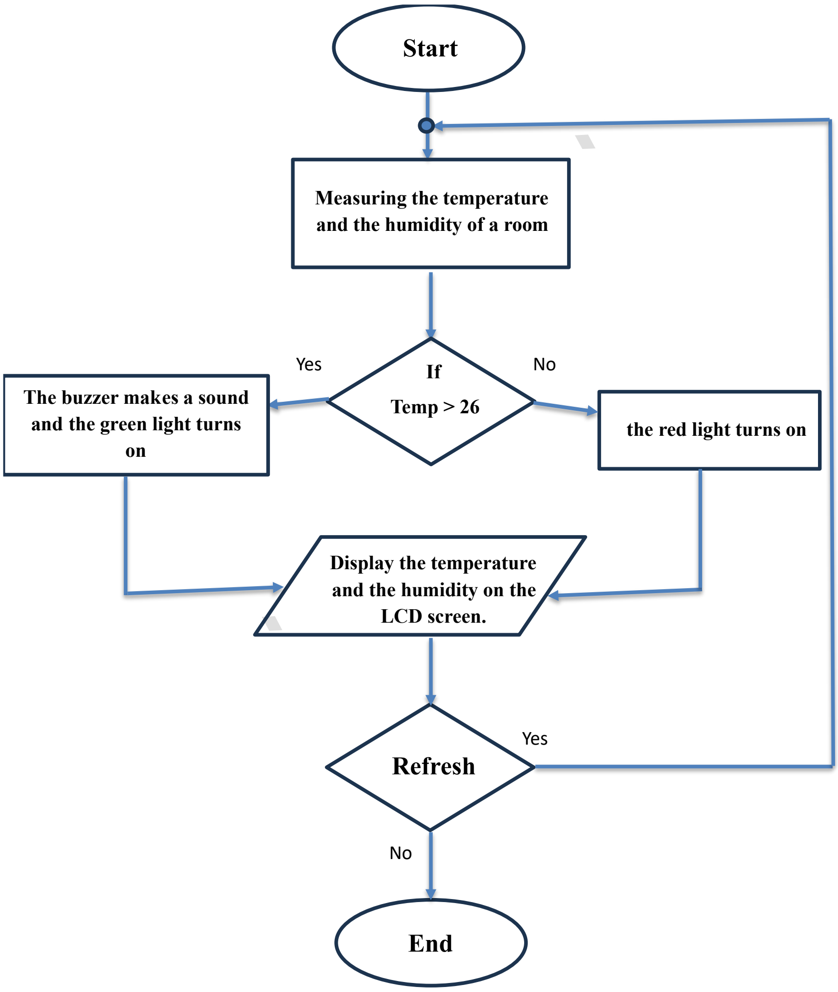


Fig (14): Flow Chart.

7. Arduino Software:

7.1. Arduino IDE

This application is compatible with multiple platforms, including Linux, macOS, and Windows. It is developed using C and C++. The Arduino Integrated Development Environment (IDE) is utilized to upload the program to the Arduino board. It accommodates both C and C++ languages, employing a unique syntax for code organization. The Arduino IDE offers a comprehensive library of programming resources. The code created by users necessitates just two fundamental functions: one to initialize the schematic and another to run the main program.

Fig (15): Arduino IDE.

7.2. the Code of the Project:

```
#include <DHT.h>
#include <LiquidCrystal_I2C.h>

int light = 8;
int light1 = 9;
int buzzer = 7;

DHT dht(2, DHT11);
int temp;
int hum;

LiquidCrystal_I2C lcd(0x27,32,4);
void setup() {
  // put your setup code here, to run once:
dht.begin();
Serial.begin(9600);
lcd.init();
lcd.backlight();

pinMode(light, OUTPUT);
pinMode(light1, OUTPUT);
pinMode(buzzer, OUTPUT);}
void loop() {
  delay(2000);
  temp = dht.readTemperature();
  hum = dht.readHumidity();

  lcd.setCursor(0,0);
  lcd.print("CPU");

  lcd.setCursor(11,0);
  lcd.print("Project 1");

  lcd.setCursor(0,2);
  lcd.print("Temperature: ");
  lcd.print(temp);
  lcd.print(" C");

  lcd.setCursor(0,3);
  lcd.print("Humidity: ");
  lcd.print(hum);
  lcd.print(" %");
```

```
if(dht.readTemperature() > 26)
 {digitalWrite(light, HIGH);
  digitalWrite(buzzer, LOW);
  digitalWrite(light1, HIGH);
 }
 else{
 digitalWrite(light1, LOW);
 digitalWrite(light, HIGH);
 digitalWrite(buzzer, LOW);}
// put your main code here, to run repeatedly:
}
```

Fig (16): Code of the Project.

8. the Implementation of the Project:

The project has been implemented, the wires have been connected to the project components, the Arduino operation has been verified, and the implementation code has been added.

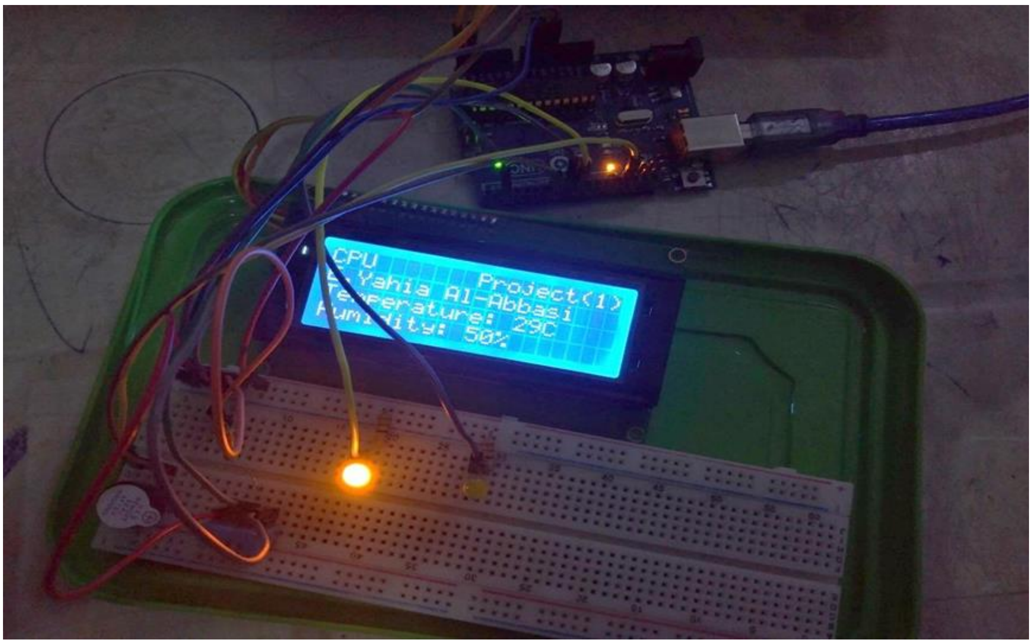
This device uses a DHT11 sensor to measure the temperature and humidity in the room and display the values on a 20x4 LCD screen via the Arduino UNO R3 board. This system is an integrated digital data processing system. The Operating Principle is:

1. System operation: When the device is connected to a power source via a USB cable or an external power source, the Arduino starts operating the sensors and electronic circuits.
2. Data reading: The DHT11 sensor measures the temperature and humidity every second and sends the values to the Arduino controller.
3. Data processing: The Arduino processes the data and sends it to the display for display.
4. Signals and alerts:

When the temperature or humidity exceeds certain limits, a buzzer is activated to provide an audible alert.

LEDs are used to indicate the environmental status; for example, green light indicates normal conditions and red light indicates critical values.

5. Data updating: Information is continuously updated to monitor changes in the surrounding environment.



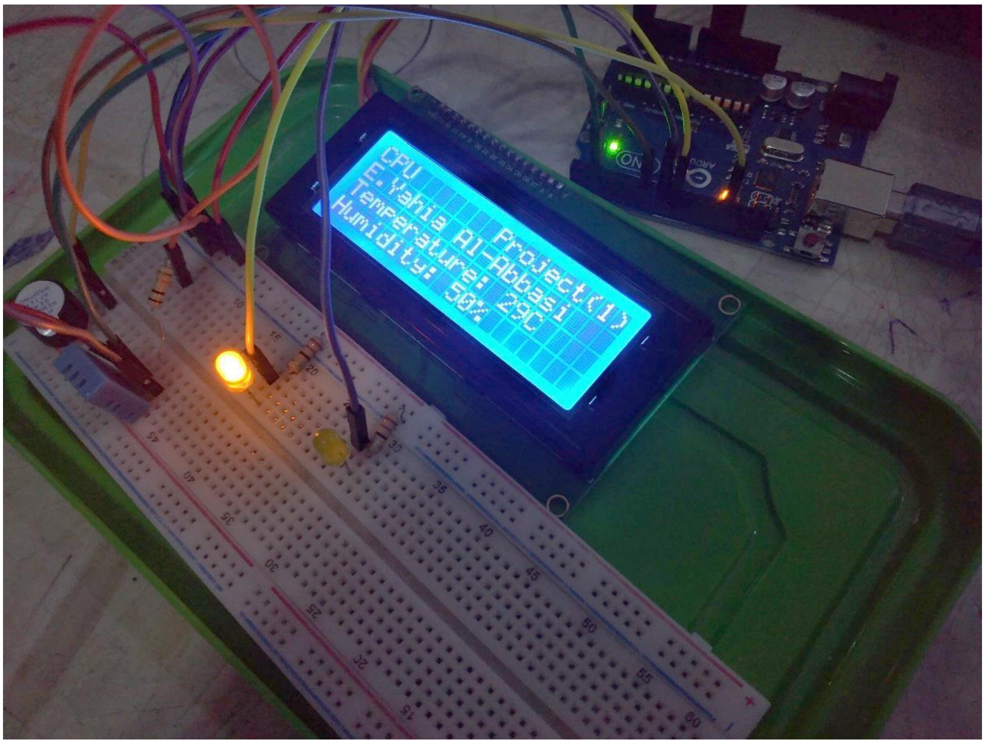


Fig (17): the Implementation of the Project.

9. Results

The implemented system successfully measured temperature and humidity in real time using an Arduino-based setup. The DHT11 sensor accurately collected environmental data, which was displayed on an LCD screen for easy monitoring. The system responded effectively to temperature variations by activating a buzzer and LED indicators based on predefined thresholds. When the temperature exceeded 26°C, the buzzer sounded, and a green light turned on, signaling high temperature. Otherwise, a red light indicated normal conditions.

This device is not only useful for home automation but also has significant healthcare applications. It helps maintain optimal environmental conditions in hospitals, clinics, and homes, reducing risks associated with respiratory diseases and heat stress. Future improvements could include wireless connectivity and data logging for better usability and remote monitoring.

10. Conclusion

This study successfully developed a temperature and humidity monitoring system using an Arduino-based setup. The system efficiently measured environmental conditions and provided real-time alerts through LED indicators and a buzzer when temperature thresholds were exceeded. The integration of an LCD screen allowed for easy data visualization, making it a user-friendly solution.

Beyond home automation, this device has significant healthcare applications, helping to maintain a healthy indoor environment in hospitals, clinics, and homes. By preventing heat stress and respiratory issues, it contributes to overall well-being. Future improvements, such as wireless connectivity and cloud-based data storage, could enhance its efficiency and usability.

References:

1. HWLibre. "Arduino LCD Screens." [Online]. Available: https://www.hwlibre.com/en/arduino-lcd-screens .
2. Wikipedia. "Arduino Overview." [Online]. Available: https://en.wikipedia.org/wiki/Arduino.

3. Menatech. "What is Arduino and its Types?" [Online]. Available: https://www.menatech.net/arduino-and-its-types .
4. Wikipedia. "Light Emitting Diode (LED)." [Online]. Available: https://en.wikipedia.org/wiki/Light-emitting_diode .
5. Electronics Go. "DHT11 Humidity Sensor with Arduino." [Online]. Available: https://electronics-go.com/dht11-humidity-sensor-arduino .
6. Arrajol. "Ideal Room Temperature for Comfort." [Online]. Available: https://www.arrajol.com/content/265076 .
7. Mawdoo3. "Humidity Measurement Device." [Online]. Available: https://www.mawdoo3.com/humidity-measurement-device .
8. Masrawy. "Measuring Air Humidity: The Importance of Hygrometer at Home." [Online]. Available: https://www.masrawy.com/howa_w_hya/health/details/2020/1/21/1709263 .
9. Wikipedia. "Electrical Resistance." [Online]. Available: https://en.wikipedia.org/wiki/Electrical_resistance .
10. Wikipedia. "Buzzer Overview." [Online]. Available: https://en.wikipedia.org/wiki/Buzzer .
11. Arduino Iraq. "Arduino IDE Overview." [Online]. Available: http://arduinoiraq.blogspot.com/2015/06/arduino-ide.html .
12. Wikipedia. "Integrated Development Environment for Arduino." [Online]. Available: https://en.wikipedia.org/wiki/Arduino_IDE .
13. Seeed Studio. "I2C LCD Guide." [Online]. Available: https://wiki.seeedstudio.com/I2C_LCD .
14. SparkFun. "Introduction to Breadboards." [Online]. Available: https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard .
15. Adafruit. "Connecting DHT11 Sensor to Arduino." [Online]. Available: https://learn.adafruit.com/dht .
16. Microchip. "Microcontroller Applications in Sensor Systems." [Online]. Available: https://www.microchip.com/en-us/applications/home-automation .